Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for generating a user interface, the user interface being configured for use in a client-server environment, the method comprising:

providing an editor for designing a visual representation of a user interface, the editor providing a workspace and a task panel to be displayed on a display device on a client system, the workspace being provided to design the visual representation thereon, the task panel providing a plurality of elements for use in designing the visual representation, one or more of the elements being associated with a server system remotely located from the client system;

selecting a first actor from the task panel, the first actor being a data source object that is one of the elements and includes application logic needed to access application layer provided in the server system;

inserting the first actor into the workspace;

inserting a second actor selected from the task panel into the workspace;

diagrammatically defining a behavioral relationship between the first actor and the second actor; and

generating executable code from the first and second actors and the behavioral relationship thereof.

2. (Currently Amended) The method of claim 1, further comprising:

generating a canonic representation of the first and second actor and the behavioral relationship thereof, wherein the executable code is generated from the canonic representation.

3. (Original) The method of claim 2, wherein the generated executable code is

compatible with a first platform, wherein the canonic representation is used to generate executable code for a second platform.

4. (Currently Amended) The method of claim 1, further comprising:

inserting an operator to the workspace, the operator being configured to process data in a specific way;

diagrammatically defining a behavioral relationship between the second actor and the operator.

5. (Original) The method of claim 1, further comprising:

storing an identifier of the first actor in a work session associated with the current instance of the editor, wherein the identifier of the first actor is used to call the first actor stored in the server system during a runtime to have the first actor perform a predetermined task.

6. (Original) The method of claim 1, further comprising:

logging on to the server system to launch the editor.

7. (Original) The method of claim 6, further comprising:

storing the generated executable code to a repository in the server system.

8. (Original) The method of claim 7, wherein the server system includes an enterprise portal, the enterprise portal including one or more servers dedicated to the application layer and one or more Web servers dedicated to interface with a plurality of client systems.

9. (Currently Amended) A method for generating a user interface using a modeling system, comprising:

providing an editor for designing a visual representation of a user interface from a server system to a client system, the editor providing a workspace and a task panel to be displayed on a display device on the client system, the workspace being provided to design the visual representation thereon, the task panel providing a plurality of elements for use in designing the visual representation, one or more of the elements being associated with the server system remotely located from the client system;

displaying a scenario selected by a user on the workspace, the scenario being compatible with user requirements for the user interface, the scenario including a plurality of interleaved scenes;

defining each of the plurality of scenes according to inputs received from the user, each scene including concurrently active and collaborating actors, the actors being specialized computational units that represent threads of activities, where each scene is defined by diagrammatically defining a behavioral relationship between the actors associated with that scene;

generating a canonic representation of a model represented by the scenario and the scenes; and

generating a first and second executable ~~of~~ from the canonic representation, wherein the first code is compatible with a first platform and the second code is compatible with a second platform that is different from the first platform.

10. (Currently Amended) A method for generating a user interface in a ~~distributed~~ computer system comprising a plurality of computers coupled in a network, the method comprising:

displaying a first business function component selected by a first user on a first display area of a frontend system, the first business function component being associated with first application logic to access a first business application provided in a server system;

displaying a second business function component selected by the first user on the first display area of the client system, the second business function component being associated with second application logic to access a second business application provided in the server system; ~~and~~

forming a behavioral relationship between the first and second business function components~~;~~;

~~wherein~~ creating a visual representation of the user interface based on the displaying steps and the forming step~~;~~;

generating a canonic representation of the visual representation; and

generating a first and second executable user interface code from the canonic representation, the first and second executable user interface code being operable to access the first and second applications provided in the server system to retrieve desired data,

wherein the first code is compatible with a first platform and the second code is compatible with a second platform that is different from the first platform.

11. (Canceled).

12. (Currently Amended) The method of claim 1~~1~~0, further comprising:

storing the user interface code in a repository in the server system, wherein the first and second business applications are different applications.

13. (Currently Amended) The method of claim 1~~1~~0, wherein the visual representation includes a third business function component that specifies a presentation format, the method further comprising:

storing the user interface code in a repository associated with the server system; and receiving a request to access the user interface code from a second user, wherein the user interface code is executed in response to the request, the code being used to access the first and second applications provided in the server

system to retrieve data desired by the second user, wherein the data retrieved for the second user is displayed on a second display area of the client system according to presentation format specified by the third business function component, the first and second display areas being associated with different client systems.

14. (Original) The method of claim 10, further comprising:

storing a first identifier for the first business function component in the client system in conjunction with the displaying-a-first-business-function-component step; and

storing a second identifier for the second business function component in the client system in conjunction with the displaying-a-second-business-function-component step,

wherein the first and second identifiers are used subsequently at a runtime to access the first and second application logics, respectively.

15. (Currently Amended) The method of claim 14~~, further comprising:~~

~~generating a canonic representation of the visual representation; and~~

~~generating a first executable user interface code from the canonic representation, the first user interface code being operable to access the first and second applications provided in the server system to retrieve desired data,~~

wherein the first and second identifiers are inserted in the canonic representation.

16. (Original) The method of claim 15, wherein the canonic representation is based on an XML-based language, wherein the first and second applications may be the same application or different applications.

17. (Canceled).

18. (Original) The method of claim 10, further comprising:

associating an operator to the second business function component; and

connecting an output port of the second business function component to an input port of the operator.

19. (Canceled).

20. (Canceled).

21. (Currently Amended) The method of claim 1~~9~~8, wherein the operator processes data received from the second business function component in such a way that the data remain consistent with the second application logic associated with the second business function component.

22. (Currently Amended) The method of claim 10, wherein the first and second business function components are ~~reusable application components~~ selected from a predefined set of business function components.

23. (Currently Amended) The method of claim 22, wherein the predefined set of business function components ~~reusable application components~~ include a business application program interface (BAPI) and a remote function call (RFC).

24. (Currently Amended) A ~~distributed~~ computer system comprising a plurality of computers coupled in a network, comprising:

means for displaying a first business function component selected by a first user on a first display area of a client system, the first business function component being associated with first application logic to access a first business application provided in a server system;

means for displaying a second business function component selected by the first user on the first display area of the client system, the second business function component being associated with second application logic to access a second business application provided in the server system; ~~and~~

means for forming a behavioral relationship between the first and second business function components, wherein a visual representation of the user interface is created based on the displaying steps and the forming step~~.~~;

means for generating a canonic representation of the visual representation; and

means for generating a first and second executable user interface code from the canonic representation, the first and second executable user interface code being operable to access the first and second applications provided in the server system to retrieve desired data,

wherein the first code is compatible with a first platform and the second code is compatible with a second platform that is different from the first platform.

25. (Canceled).

26. (Currently Amended) The system of claim 2~~5~~4, further comprising: means for storing the user interface code in a repository in the server system.

27. (Original) The system of claim 24 wherein the visual representation includes a third business function component that specifies a presentation format, the system further comprising:

means for storing the user interface code in a repository associated with the server system; and

means for receiving a request to access the user interface code from a second user,

wherein the user interface code is executed in response to the request, the code being used to access the first and second applications provided in the server system to retrieve data desired by the second user,

wherein the data retrieved for the second user is displayed on a second display area of the client system according to presentation format specified by the third business function component, the first and second display areas being associated with different client systems.

28. (Currently Amended) A computer readable medium including a computer program, the computer program including:

code for displaying a first business function component selected by a first user on a first display area of a client system, the first business function component being associated with first application logic to access a first business application provided in a server system;

code for displaying a second business function component selected by the first user on the first display area of the client system, the second business function component being associated with second application logic to access a second business application provided in the server system; ~~and~~

code for forming a behavioral relationship between the first and second business function components, wherein a visual representation of the user interface is created based on the displaying steps and the forming step~~.~~;

code for generating a canonic representation of the visual representation; and

code for generating a first and second executable user interface code from the canonic representation, the first and second executable user interface code being operable to access the first and second applications provided in the server system to retrieve desired data,

wherein the first code is compatible with a first platform and the second code is compatible with a second platform that is different from the first platform.

29. (Canceled).

30. (Currently Amended) A computer system, comprising:

applications provided on a server system coupled to a client system; and

a computer readable medium including:

code for displaying a first business function component selected by a first user on a first display area of the client system, the first business function component being associated with first application logic to access one or more business applications provided in a server system;

code for displaying a second business function component selected by the first user on the first display area of the client system, the second business function component being associated with second application logic to access one or more business applications provided in the server system;~~ and~~

code for forming a behavioral relationship between the first and second business function components;; and

~~wherein a visual representation of the user interface is created based on the displaying steps and the forming step.~~

code for generating a first and second executable user interface code from the displaying steps and the forming step, the first and second executable user interface code being operable to access the first and second applications provided in the server system to retrieve desired data,

wherein the first code is compatible with a first platform and the second code is compatible with a second platform that is different from the first platform.